

第5章 数组和广义表

DATA STRUCTURE

计算机科学学院 廖雪花

本章内容简介

数组和广义表

5.1 数组的定义

5.2 数组的顺序表示和实现

5.3 矩阵的压缩存储

5.4 广义表的定义

5.5 广义表的存储结构

5.3 矩阵的压缩存储

廖雪花 LiaoXuehua

5.3.2 稀疏矩阵

- ◆ 几种压缩存储方案

- 三元组顺序表

- 带行逻辑链接的顺序表

- 十字链表

二、带行逻辑链接的顺序表

■ 引入

	0	12	9	0	0	0	0
	0	0	0	0	0	0	0
M=	-3	0	0	0	0	14	0
	0	0	24	0	0	0	0
	0	18	0	0	0	0	0
	15	0	0	-7	0	0	0

M.data[1]	<i>i</i>	<i>j</i>	<i>e</i>
	1	2	12
	1	3	9
	3	1	-3
	3	6	14
	4	3	24
	5	2	18
	6	1	15
	6	4	-7

M.mu=6, M.nu=7, M.tu=8

行逻辑链接的顺序表存储定义

```
typedef struct{
    Triple data[MAXSIZE+1]; //非0元三元组表
    int rpos[MAXC+1]; //各行第一个非0元的位置表
    int mu,nu,tu; //矩阵的行数、列数、非0元个数
}RLSMatrix
```

例如：稀疏矩阵的行逻辑链接顺序表

$$M = \begin{vmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{vmatrix}$$

<i>row</i>	1	2	33	44	5	6
<i>num[row]</i>	2	0	2	1	1	2
<i>M.rpos[row]</i>	1	3	3	5	6	7

一般地：

$$\left\{ \begin{array}{l} M.rpos[1] = 1 \\ M.rpos[row] = M.rpos[row - 1] + num[row - 1] \quad 2 \leq row \leq M.mu \end{array} \right.$$

二、带行逻辑链接的顺序表

◆ 思考：第row行元素在M.data表中的访问起止位置？

□ 起始位置：

- $M.rpos[row]$

□ 终止位置：

- $M.rpos[row + 1] - 1 \quad (1 \leq row \leq M.mu - 1)$
- $M.tu \quad (row = M.mu)$

稀疏矩阵的乘积运算

■ 分析：

- ◆ M的每个非0元M.data[p] (i, k, M_{ik}) ($p = 1, 2, \dots, M.tu$) 与N的所有相应元素 N.data[q] (k, j, M_{kj}) ($q = 1, 2, \dots, N.tu$) 相乘并累计求和即可。

$$Q_{ij} = \sum_{k=1}^{M.nu} M_{ik} \times N_{kj} (1 \leq i \leq M.mu, 1 \leq j \leq N.nu)$$

- ◆ 累乘的结果可以先暂时存入一个临时数组ctemp中。
- ◆ 两个稀疏矩阵的乘积不一定为稀疏矩阵。

稀疏矩阵的乘积运算

```
Status MultSMatrix(RLMatrix M,RLMatrix N,RLMatrix &Q){  
    if (M.nu!=N.mu) return ERROR;  
    Q.mu=M.mu;Q.nu=N.nu;Q.tu=0;  
    if( M.tu*N.tu!=0) {  
        for (arow=1;arow<=M.mu;arow++) {  
            ctemp[ ]=0;  Q.rpos[arow]=Q.tu+1;  
            if (arow<M.tu) tp=M.rpos[arow+1]; else tp=M.tu+1;  
            for (p=M.rpos[arow];p<tp ;++p) {  
                brow=M.data[p].j;  
                if (brow<N.nu) t=N.rpos[brow+1];    else t=N.tu+1;  
                for (q=N.rpos[brow];q<t;++q) {  
                    ccol=N.data[q].j;  ctemp[ccol]+=M.data[p].e*N.data[q].e;  
                } //for q  
            } //for p  
    } //for Q
```

稀疏矩阵的乘积运算

```
for (ccol=1; ccol<=Q.nu; ++ccol)
    if (ctemp[ccol]) {
        if (++Q.tu>MAXSIZE) return ERROR;
        Q.data[Q.tu] = {arow, ccol, ctemp[ccol]};
    } //if
} //for arow
} //if
return OK;
} //MultSMatrix
```

本节要点

- 稀疏矩阵的压缩存储：
 - ✓ 稀疏矩阵的定
 - ✓ 稀疏矩阵的存储：三元组顺序表、带行逻辑链接的顺序表、十字链表
- 带行逻辑链接的顺序表：
 - ✓ 带行逻辑链接的顺序表存储表示
 - ✓ 矩阵的乘法

感谢聆听